

1、关于linux

1.1 为什么要使用linux

我自己是这样理解的，在Java开发的过程中我们需要将自己的项目部署到服务器（本人使用的是阿里云服务器）中去，而我们选择的服务器linux操作系统，我们要想去使用这些服务器我们就要懂得linux一些简单使用，至少简单的指令、在服务器安装环境以及怎么将项目部署到阿里云上让大家都可以访问到，这些是我们必须要会的。

1.2 linux概述

1. Linux 内核最初只是由芬兰人林纳斯·托瓦兹（Linus Torvalds）在赫尔辛基大学上学时出于个人爱好而编写的。
2. Linux 是一套免费使用和自由传播的类 Unix 操作系统，是一个基于 POSIX（可移植操作系统接口）和 UNIX 的多用户、多任务、支持多线程和多 CPU 的操作系统。
3. Linux 能运行主要的 UNIX 工具软件、应用程序和网络协议。它支持 32 位和 64 位硬件。Linux 继承了 Unix 以网络为核心的设计思想，**是一个性能稳定的多用户网络操作系统**。
4. 目前市面上较知名的发行版有：Ubuntu、RedHat、CentOS、Debian、Fedora、SuSE、OpenSUSE、Arch Linux、SolusOS 等，我在学习的过程中使用的是CentOS7.7版本。
5. 今天各种场合都有使用各种 Linux 发行版，从嵌入式设备到超级计算机，并且在服务器领域确定了地位，通常服务器使用 LAMP（Linux + Apache + MySQL + PHP）或 LNMP（Linux + Nginx + MySQL + PHP）组合。

1.3 linux和windows的比较

比较	Windows	Linux
界面	界面统一，外壳程序固定所有 Windows 程序菜单几乎一致，快捷键也几乎相同	图形界面风格依发布版不同而不同，可能互不兼容。GNU/Linux 的终端机是从 UNIX 传承下来，基本命令和操作方法也几乎一致。
驱动程序	驱动程序丰富，版本更新频繁。默认安装程序里面一般包含有该版本发布时流行的硬件驱动程序，之后所出的新硬件驱动依赖于硬件厂商提供。对于一些老硬件，如果没有了原配的驱动有时很难支持。另外，有时硬件厂商未提供所需版本的 Windows 下的驱动，也会比较头痛。	由志愿者开发，由 Linux 核心开发小组发布，很多硬件厂商基于版权考虑并未提供驱动程序，尽管多数无需手动安装，但是涉及安装则相对复杂，使得新用户面对驱动程序问题（是否存在和安装方法）会一筹莫展。但是在开源开发模式下，许多老硬件尽管在 Windows 下很难支持的也容易找到驱动。HP、Intel、AMD 等硬件厂商逐步不同程度支持开源驱动，问题正在得到缓解。
使用	使用比较简单，容易入门。图形化界面对没有计算机背景知识的用户使用十分有利。	图形界面使用简单，容易入门。文字界面，需要学习才能掌握。
学习	系统构造复杂、变化频繁，且知识、技能淘汰快，深入学习困难。	系统构造简单、稳定，且知识、技能传承性好，深入学习相对容易。
软件	每一种特定功能可能都需要商业软件的支持，需要购买相应的授权。	大部分软件都可以自由获取，同样功能的软件选择较少。

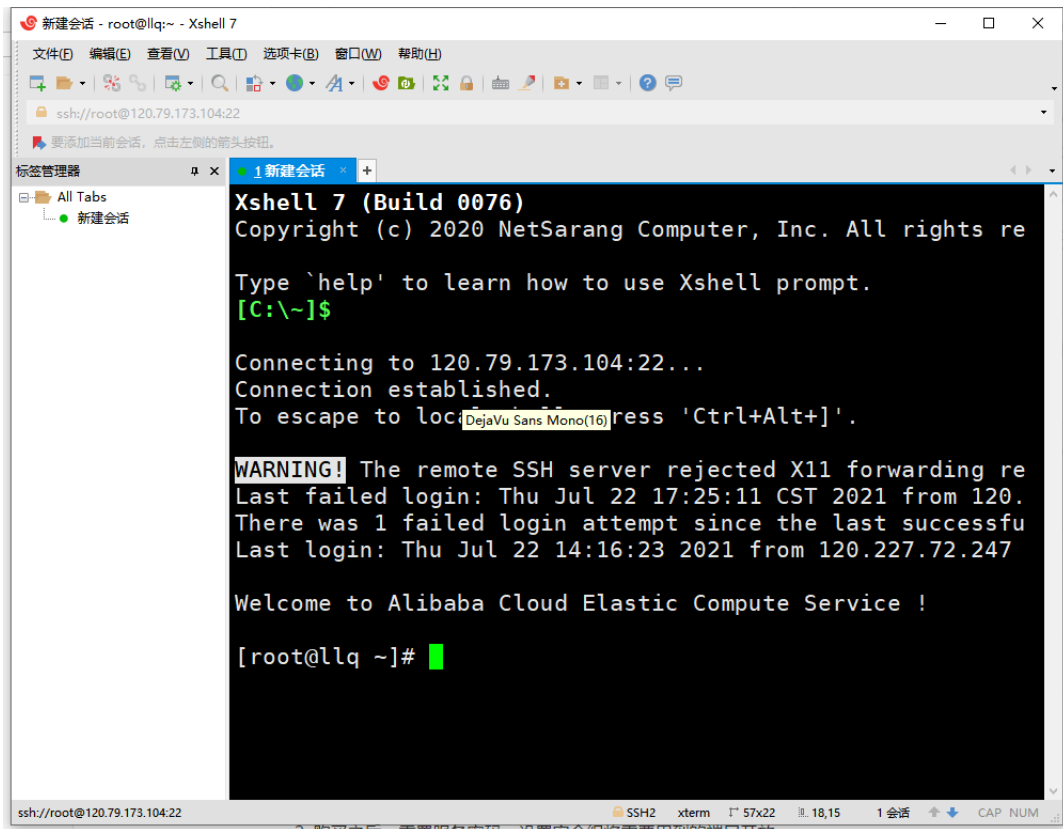
狂神说

1.4 linux环境的搭建

1. 安装CentOS虚拟机（不推荐）

2. 直接购买一个云服务器（个人采用这种方式）

1. 购买阿里云服务器：<https://www.aliyun.com/minisite/goods?userCode=0phtycgr>
2. 购买之后，重置服务密码，设置安全组将需要用到的端口开放。
3. 下载xshell工具，输入公网ip和密码进行远程连接使用，连接成功的界面如下图所示



注意事项：

如果要打开端口，需要在阿里云的安全组面板中开启对应的出入规则，不然的话会被阿里拦截！

如果前期不好操作，可以推荐安装宝塔面板，傻瓜式管理服务器（阿里云服务器入门文章里面有简述）

2、linux的基本指令

2.1 开机关机登录

1. 开机启动许多程序。它们在Windows叫做"服务" (service)，在Linux就叫做"守护进程" (daemon)。

开机成功后，它会显示一个文本登录界面，这个界面就是我们经常看到的登录界面，在这个登录界面中会提示用户输入用户名，而用户输入的用户将作为参数传给login程序来验证用户的身份，linux中密码是不显示的，输完回车即可！

一般来说，用户的登录方式有三种：

1. 命令行登录
2. ssh登录
3. 图形界面登录

2.关机指令（一般用不到）

- 1 sync # 将数据由内存同步到硬盘中。
- 2

```
3 shutdown # 关机指令，你可以man shutdown 来看一下帮助文档。例如你可以运行如下命令关机：
4
5 shutdown -h 10 # 这个命令告诉大家，计算机将在10分钟后关机
6
7 shutdown -h now # 立马关机
8
9 shutdown -h 20:25 # 系统会在今天20:25关机
10
11 shutdown -h +10 # 十分钟后关机
12
13 shutdown -r now # 系统立马重启
14
15 shutdown -r +10 # 系统十分钟后重启
16
17 reboot # 就是重启，等同于 shutdown -r now
18
19 halt # 关闭系统，等同于shutdown -h now 和 poweroff
```

不管是重启系统还是关闭系统，首先要运行 **sync** 命令，把内存中的数据写到磁盘中。

2.2 处理目录指令以及系统中各文件夹的解释

```
1 ls: 列出目录
2
3 cd: 切换目录
4
5 pwd: 显示目前的目录
6
7 mkdir: 创建一个新的目录
8
9 rmdir: 删除一个空的目录
10
11 cp: 复制文件或目录
12
13 rm: 移除文件或目录
14
15 mv: 移动文件与目录，或修改文件与目录的名称
```

1. ls指令

作用：列出当前目录下的可见文件以及文件夹

```
1 [root@llq ~]# cd /
2 [root@llq /]# ls
3 bin    home      lost+found patch  sbin    usr
4 boot   install.sh media      proc   srv     var
5 dev    lib       mnt       root   sys     www
6 etc    lib64     opt       run    tmp
```

所有的目录都挂载在/根目录下，以下是对于这些目录的解释：

```
1 /bin: bin是Binary的缩写，这个目录存放着最经常使用的命令。
2
3 /boot: 这里存放的是启动Linux时使用的一些核心文件，包括一些连接文件以及镜像文件。
4
```

5 /dev : dev是Device(设备)的缩写, 存放的是Linux的外部设备, 在Linux中访问设备的方式和访问文件的方式是相同的。

6

7 /etc: 这个目录用来存放所有的系统管理所需要的配置文件和子目录。

8

9 /home: 用户的主目录, 在Linux中, 每个用户都有一个自己的目录, 一般该目录名是以用户的账号命名的。

10

11 /lib: 这个目录里存放着系统最基本的动态连接共享库, 其作用类似于windows里的DLL文件。

12

13 /lost+found: 这个目录一般情况下是空的, 当系统非法关机后, 这里就存放了一些文件。

14

15 /media: linux系统会自动识别一些设备, 例如U盘、光驱等等, 当识别后, linux会把识别的设备挂载到这个目录下。

16

17 /mnt: 系统提供该目录是为了让用户临时挂载别的文件系统的, 我们可以将光驱挂载在/mnt/上, 然后进入该目录就可以查看光驱里的内容了。

18

19 /opt: 这是给主机额外安装软件所摆放的目录。比如你安装一个ORACLE数据库则就可以放到这个目录下。默认是空的。

20

21 /proc: 这个目录是一个虚拟的目录, 它是系统内存的映射, 我们可以通过直接访问这个目录来获取系统信息。

22

23 /root: 该目录为系统管理员, 也称作超级权限者的用户主目录。

24

25 /sbin: s就是Super User的意思, 这里存放的是系统管理员使用的系统管理程序。

26

27 /srv: 该目录存放一些服务启动之后需要提取的数据。

28

29 /sys: 这是linux2.6内核的一个很大的变化。该目录下安装了2.6内核中新出现的一个文件系统sysfs。

30

31 /tmp: 这个目录是用来存放一些临时文件的。

32

33 /usr: 这是一个非常重要的目录, 用户的很多应用程序和文件都放在这个目录下, 类似于windows下的program files目录。

34

35 /usr/bin: 系统用户使用的应用程序。

36

37 /usr/sbin: 超级用户使用的比较高级的管理程序和系统守护程序。

38

39 /usr/src: 内核源代码默认的放置目录。

40

41 /var: 这个目录中存放着在不断扩充着的东西, 我们习惯将那些经常被修改的目录放在这个目录下。包括各种日志文件。

42

43 /run: 是一个临时文件系统, 存储系统启动以来的信息。当系统重启时, 这个目录下的文件应该被删掉或清除。

44

45 /www: 一般宝塔自动安装的软件会存放在这个下面的server目录

2. mkdir建立一个新目录

```
1 | mkdir [-mp] 目录名称
```

参数:

- 1 `-m` : 配置文件的权限喔! 直接配置, 不需要看默认权限 (`umask`) 的脸色~
- 2 `-p` : 帮助你直接将所需要的目录(包含上一级目录)递归创建起来!

示例:

```
1 #创建多层目录
2 [root@llq home]# ls
3 lilingqi llq.txt redis test1 www
4 [root@llq home]# mkdir test2/test3/test4
5 mkdir: cannot create directory 'test2/test3/test4': No such file or
  directory //这样不能多层创建
6 [root@llq home]# mkdir -p test2/test3/test4 //需要加上-p才能创建成功
7 [root@llq home]# ls
8 lilingqi llq.txt redis test1 test2 www
9 [root@llq home]# cd test2
10 [root@llq test2]# ls
11 test3 //这个里面还有一个test4
12 #创建权限为 rwxrwxrwx的test5文件夹
13 [root@llq home]# ls
14 lilingqi llq.txt redis test1 test2 www
15 [root@llq home]# mkdir -m 777 test5
16 [root@llq home]# ls -l
17 total 28
18 drwx----- 2 lilingqi lilingqi 4096 Jul 22 11:08 lilingqi
19 -rw-r--r-- 1 root      root      20 Jul 21 20:44 llq.txt
20 drwx----- 2 redis     redis    4096 Jul 20 20:30 redis
21 drwxr-xr-x 2 root      root      4096 Jul 22 18:00 test1
22 drwxr-xr-x 3 root      root      4096 Jul 22 18:03 test2
23 drwxrwxrwx 2 root      root      4096 Jul 22 18:14 test5
24 drwx----- 3 www       www      4096 Jul 20 20:28 www
25 [root@llq home]# ^C
26 [root@llq home]#
```

3. `rmdir`: 删除一个空的目录

语法:

```
1 rmdir [-p] 目录名称
```

参数:

```
1 选项与参数: -p : 连同下一级『空的』目录也一起删除
```

注意: 只能删除空目录, 删除非空利用`rm`指令

4. `cp`: 复制文件和目录

语法:

```
1 [root@www ~]# cp [-adfilprsu] 来源档(source) 目标档(destination)
2 [root@www ~]# cp [options] source1 source2 source3 .... directory
```

选项和参数

```

1 a: 相当于 -pdr 的意思, 至于 pdr 请参考下列说明; (常用)
2
3 -p: 连同文件的属性一起复制过去, 而非使用默认属性(备份常用);
4
5 -d: 若来源档为连结档的属性(link file), 则复制连结档属性而非文件本身;
6
7 -r: 递归持续复制, 用于目录的复制行为; (常用)
8
9 -f: 为强制(force)的意思, 若目标文件已经存在且无法开启, 则移除后再尝试一次;
10
11 -i: 若目标档(destination)已经存在时, 在覆盖时会先询问动作的进行(常用)
12
13 -l: 进行硬式连结(hard link)的连结档创建, 而非复制文件本身。
14
15 -s: 复制成为符号连结档 (symbolic link), 亦即『捷径』文件;
16
17 -u: 若 destination 比 source 旧才升级 destination !

```

示例:

```

1 # 找一个有文件的目录, 我这里找到 root目录
2 [root@llq home]# cd /root
3 [root@llq ~]# ls
4 install.sh
5 [root@kuangshen ~]# cd /home
6
7 # 复制 root目录下的install.sh 到 home目录下
8 [root@llq home]# cp /root/install.sh /home
9 [root@llq home]# ls
10 install.sh
11
12 # 再次复制, 加上-i参数, 增加覆盖询问?
13 [root@llq home]# cp -i /root/install.sh /home
14 cp: overwrite '/home/install.sh'? y # n不覆盖, y为覆盖

```

5. rm: 移除文件或者目录

语法:

```

1 rm [-fir] 文件或目录

```

选项和参数

```

1 -f : 就是 force 的意思, 忽略不存在的文件, 不会出现警告信息;
2
3 -i : 互动模式, 在删除前会询问使用者是否动作
4
5 -r : 递归删除! 最常用在目录的删除了! 非常危险的选项!!!

```

示例:

```

1 #可以这样来删除多层目录
2 root@llq home]# ls
3 lilingqi llq.txt redis test1 www
4 [root@llq home]# rm -r test1 #加 -r

```

```

5  rm: descend into directory 'test1'? y
6  rm: remove directory 'test1/test2'? y
7  rm: remove directory 'test1'? y
8  [root@llq home]# ls
9  lilingqi llq.txt redis www
10 #在删除的时候进行询问
11 [root@llq home]# ls
12 lilingqi llq.txt redis www
13 [root@llq home]# rm -i llq.txt #加-i
14 rm: remove regular file 'llq.txt'? y
15 [root@llq home]# ls
16 lilingqi redis www
17 [root@llq home]#

```

6. mv: 移动目录或者文件, 也可以用来修改文件名

语法:

```

1 [root@www ~]# mv [-fiu] source destination
2 [root@www ~]# mv [options] source1 source2 source3 .... directory

```

选项和参数:

```

1 -f : force 强制的意思, 如果目标文件已经存在, 不会询问而直接覆盖;
2
3 -i : 若目标文件 (destination) 已经存在时, 就会询问是否覆盖!
4
5 -u : 若目标文件已经存在, 且 source 比较新, 才会升级 (update)

```

示例:

```

1 #移动文件
2 [root@llq home]# ls
3 lilingqi llq.txt redis www
4 [root@llq home]# mv llq.txt lilingqi
5 [root@llq home]# ls
6 lilingqi redis www
7 [root@llq home]# cd lilingqi
8 [root@llq lilingqi]# ls
9 apache-tomcat-8.0.50-windows-x64.zip
10 jdk-8u301-linux-x64.rpm
11 linux_test-0.0.1-SNAPSHOT.jar
12 llq.txt
13 #改名
14 [root@llq home]# ls
15 lilingqi redis www
16 [root@llq home]# mv lilingqi Lilingqi
17 [root@llq home]# ls
18 Lilingqi redis www
19 [root@llq home]# mv Lilingqi lilingqi
20 [root@llq home]# ls
21 lilingqi redis www
22 [root@llq home]# mv lilingqi SB #将lilingqi变成SB
23 [root@llq home]# ls
24 redis SB www

```


2.3 怎么看懂文件的基本属性

Linux系统是一种典型的多用户系统，不同的用户处于不同的地位，拥有不同的权限。为了保护系统的安全性，Linux系统对不同的用户访问同一文件（包括目录文件）的权限做了不同的规定。

在Linux中我们可以使用 `ll` 或者 `ls -l` 命令来显示一个文件的属性以及文件所属的用户和组，如：

```
1 [root@llq /]# ls -l
2 total 96
3 lrwxrwxrwx. 1 root root 7 Apr 26 2020 bin -> usr/bin
4 dr-xr-xr-x. 5 root root 4096 Apr 26 2020 boot
5 drwxr-xr-x 19 root root 2980 Jul 20 17:56 dev
6 drwxr-xr-x. 84 root root 4096 Jul 22 14:45 etc
7 drwxr-xr-x. 5 root root 4096 Jul 22 18:45 home
8 -rw-r--r-- 1 root root 25958 Jul 22 10:36 install.sh
9 lrwxrwxrwx. 1 root root 7 Apr 26 2020 lib -> usr/lib
10 lrwxrwxrwx. 1 root root 9 Apr 26 2020 lib64 -> usr/lib64
11 drwx----- 2 root root 16384 Apr 26 2020 lost+found
12 drwxr-xr-x. 2 root root 4096 Apr 11 2018 media
13 drwxr-xr-x. 2 root root 4096 Apr 11 2018 mnt
14 drwxr-xr-x. 4 root root 4096 Jul 22 14:30 opt
15 drwxr-xr-x 2 root root 4096 Jul 20 20:18 patch
16 dr-xr-xr-x 110 root root 0 Jul 20 17:54 proc
17 dr-xr-x--- 7 root root 4096 Jul 22 18:47 root
18 drwxr-xr-x 27 root root 780 Jul 22 14:46 run
19 lrwxrwxrwx. 1 root root 8 Apr 26 2020 sbin -> usr/sbin
20 drwxr-xr-x. 2 root root 4096 Apr 11 2018 srv
21 dr-xr-xr-x 13 root root 0 Jul 21 15:38 sys
22 drwxrwxrwt. 15 root root 4096 Jul 22 15:54 tmp
23 drwxr-xr-x. 14 root root 4096 Dec 13 2016 usr
24 drwxr-xr-x. 19 root root 4096 Jul 20 20:16 var
25 drwxr-xr-x 6 root root 4096 Jul 20 20:06 www
```

- 1 实例中，`boot`文件的第一个属性用"`d`"表示。"`d`"在Linux中代表该文件是一个目录文件。
- 2 在Linux中第一个字符代表这个文件是目录、文件或链接文件等等：
- 3
- 4 当为[`d`]则是目录
- 5
- 6 当为[`-`]则是文件；
- 7
- 8 若是[`l`]则表示为链接文档（link file）；
- 9
- 10 若是[`b`]则表示为装置文件里面的可供储存的接口设备（可随机存取装置）；
- 11
- 12 若是[`c`]则表示为装置文件里面的串行端口设备，例如键盘、鼠标（一次性读取装置）。

接下来的字符中，以三个为一组，且均为『`rwX`』的三个参数的组合。

其中，[`r`]代表可读(read)、[`w`]代表可写(write)、[`x`]代表可执行(execute)。

要注意的是，这三个权限的位置不会改变，如果没有权限，就会出现减号[`-`]而已。

每个文件的属性由左边第一部分的10个字符来确定（如下图）：

文件 类型	属主 权限			属组 权限			其他用户 权限		
0	1	2	3	4	5	6	7	8	9
d	rwX			r-X			r-X		
目录 文件	读	写	执行	读	写	执行	读	写	执行

从左至右用0-9这些数字来表示。

第0位确定文件类型，第1-3位确定属主（该文件的所有者）拥有该文件的权限。第4-6位确定属组（所有者的同组用户）拥有该文件的权限，第7-9位确定其他用户拥有该文件的权限。

其中：

第1、4、7位表示读权限，如果用“r”字符表示，则有读权限，如果用“-”字符表示，则没有读权限；

第2、5、8位表示写权限，如果用“w”字符表示，则有写权限，如果用“-”字符表示没有写权限；

第3、6、9位表示可执行权限，如果用“x”字符表示，则有执行权限，如果用“-”字符表示，则没有执行权限。

对于文件来说，它都有一个特定的所有者，也就是对该文件具有所有权的用户。

同时，在Linux系统中，用户是按组分类的，一个用户属于一个或多个组。

文件所有者以外的用户又可以分为文件所有者的同组用户和其他用户。

因此，Linux系统按文件所有者、文件所有者同组用户和其他用户来规定了不同的文件访问权限。

2.4 修改文件属性

1. chgrp: 更改文件属组]

语法：

```
1 | chgrp [-R] 属组名 文件名
```

参数：

```
1 | -R: 递归更改文件属组，就是在更改某个目录文件的属组时，如果加上-R的参数，那么该目录下的所有文件的属组都会更改。
```

2. chown:更改文件属主，也可以同时改变文件属组

语法：

```
1 | chown [-R] 属主名 文件名
2 | chown [-R] 属主名:属组名 文件名
```

3. chmod: 更改文件的9个属性

语法:

```
1 | chmod [-R] xyz 文件或目录
```

示例:

```
1 | #将某文件的属主, 属组。其他用户的权限应得改成 rwxrwx---
2 | chmod 770 文件名
```

2.5 文件内容查看

2.5.1 基本指令总结

```
1 | cat 由第一行开始显示文件内容
2 |
3 | tac 从最后一行开始显示, 可以看出 tac 是 cat 的倒着写!
4 |
5 | nl 显示的时候, 顺道输出行号!
6 |
7 | more 一页一页的显示文件内容
8 |
9 | less 与 more 类似, 但是比 more 更好的是, 他可以往前翻页!
10 |
11 | head 只看头几行
12 |
13 | tail 只看尾巴几行
```

2.5.2 指令详解

1. cat 由第一行显示内容

语法:

```
1 | cat [-AbETv]
```

参数:

```
1 | -A : 相当於 -vET 的整合选项, 可列出一些特殊字符而不是空白而已;
2 |
3 | -b : 列出行号, 仅针对非空白行做行号显示, 空白行不标行号!
4 |
5 | -E : 将结尾的断行字节 $ 显示出来;
6 |
7 | -n : 列印出行号, 连同空白行也会有行号, 与 -b 的选项不同;
8 |
9 | -T : 将 [tab] 按键以 ^I 显示出来;
10 |
11 | -v : 列出一些看不出来的特殊字符
```

2. tac 跟cat相反从最后一行开始显示

3. nl 显示内容的同时显示行号

语法:

```
1 | nl [-bnw] 文件
```

参数:

```
1 | -b : 指定行号指定的方式,主要有两种: -b a : 表示不论是否为空行,也同样列出行号(类似 cat -n); -b t : 如果有空行,空的那一行不要列出行号(默认值);
2 |
3 | -n : 列出行号表示的方法,主要有三种: -n ln : 行号在荧幕的最左方显示; -n rn : 行号在自己栏位的最右方显示,且不加 0 ; -n rz : 行号在自己栏位的最右方显示,且加 0 ;
4 |
5 | -w : 行号栏位的占用的位数。
```

4. more: 显示内容的同时可以一页一页翻动

在使用more的时候可以用的一些键:

```
1 | 空白键 (space): 代表向下翻一页;
2 |
3 | Enter: 代表向下翻『一行』;
4 |
5 | /字串: 代表在这个显示的内容当中,向下搜寻『字串』这个关键字;
6 |
7 | :f: 立刻显示出档名以及目前显示的行数;
8 |
9 | q: 代表立刻离开 more , 不再显示该文件内容。
10 |
11 | b 或 [ctrl]-b : 代表来回翻页,不过这动作只对文件有用,对管线无用。
```

5. less: 跟more不同的是多了一点点内容

可以使用的功能键

```
1 | 空白键 : 向下翻动一页;
2 |
3 | [pagedown]: 向下翻动一页;
4 |
5 | [pageup] : 向上翻动一页;
6 |
7 | /字串 : 向下搜寻『字串』的功能;
8 |
9 | ?字串 : 向上搜寻『字串』的功能;
10 |
11 | n : 重复前一个搜寻 (与 / 或 ? 有关!)
12 |
13 | N : 反向的重复前一个搜寻 (与 / 或 ? 有关!)
14 |
15 | q : 离开 less 这个程序;
```

6. head: 取出文件的前面几行

语法:

```
1 | head [-n number] 文件
```

选项与参数：-n 后面接数字，代表显示几行的意思！

默认的情况中，显示前面 10 行！若要显示前 20 行，就得要这样：

```
1 | [root@llq etc]# head -n 20 /etc/csh.login
```

7. tail 取出文件后面几行

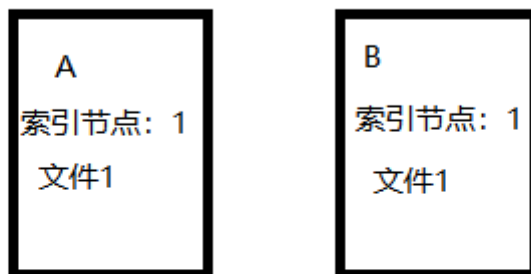
语法默认情况同上！！

3、linux链接概念

3.1 硬连接

硬连接指通过索引节点来进行连接。在 Linux 的文件系统中，保存在磁盘分区中的文件不管是什么类型都给它分配一个编号，称为索引节点号(Inode Index)。在 Linux 中，多个文件名指向同一索引节点是存在的。比如：A 是 B 的硬链接（A 和 B 都是文件名），则 A 的目录项中的 inode 节点号与 B 的目录项中的 inode 节点号相同，即一个 inode 节点对应两个不同的文件名，两个文件名指向同一个文件，A 和 B 对文件系统来说是完全平等的。删除其中任何一个都不会影响另外一个的访问。

硬连接的作用是允许一个文件拥有多个有效路径名，这样用户就可以建立硬连接到重要文件，以防止“误删”的功能。其原因如上所述，因为对应该目录的索引节点有一个以上的连接。只删除一个连接并不影响索引节点本身和其它的连接，只有当最后一个连接被删除后，文件的数据块及目录的连接才会被释放。也就是说，文件真正删除的条件是与之相关的所有硬连接文件均被删除。

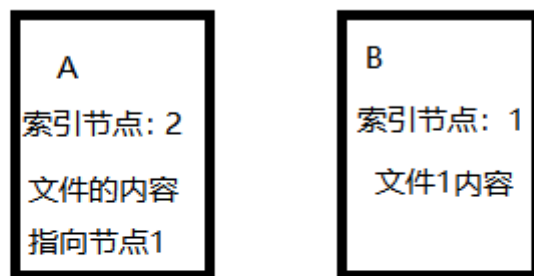


假设A是B的硬连接，那么AB中的索引节点和指向的文件都是一样的，删除中间任意一个都不会有影响类似于备份。

3.2 软连接

另外一种连接称之为符号连接（Symbolic Link），也叫软连接。软链接文件有类似于 Windows 的快捷方式。它实际上是一个特殊的文件。在符号连接中，文件实际上是一个文本文件，其中包含的有另一文件的位置信息。比如：A 是 B 的软链接（A 和 B 都是文件名），A 的目录项中的 inode 节点号与 B 的目录项中的 inode 节点号不相同，A 和 B 指向的是两个不同的 inode，继而指向两块不同的数据块。但是 A 的数据块中存放的只是 B 的路径名（可以根据这个找到 B 的目录项）。A 和 B 之间是“主从”关系，

如果 B 被删除了，A 仍然存在（因为两个是不同的文件），但指向的是一个无效的链接。



假设A是B的软连接,A中的节点和B中的节点是不一样的
但是A中的文件内容指向的是B的节点1，所以访问A的时候
可以看到文件1的内容。删除文件 B的时候，A不会消失
但是访问A的时候，已经看不到B的内容了

测试：

```
1 [root@kuangshen /]# cd /home
2 [root@kuangshen home]# touch f1 # 创建一个测试文件f1
3 [root@kuangshen home]# ls
4 f1
5 [root@kuangshen home]# ln f1 f2 # 创建f1的一个硬连接文件f2
6 [root@kuangshen home]# ln -s f1 f3 # 创建f1的一个符号连接文件f3
7 [root@kuangshen home]# ls -li # -i参数显示文件的inode节点信息
8 397247 -rw-r--r-- 2 root root 0 Mar 13 00:50 f1
9 397247 -rw-r--r-- 2 root root 0 Mar 13 00:50 f2
10 397248 lrwxrwxrwx 1 root root 2 Mar 13 00:50 f3 -> f1
```

从上面的结果中可以看出，硬连接文件 f2 与原文件 f1 的 inode 节点相同，均为 397247，然而符号（软）连接文件的 inode 节点不同。

```
1 # echo 字符串输出 >> f1 输出到 f1文件
2 [root@kuangshen home]# echo "I am f1 file" >>f1
3 [root@kuangshen home]# cat f1
4 I am f1 file
5 [root@kuangshen home]# cat f2
6 I am f1 file
7 [root@kuangshen home]# cat f3
8 I am f1 file
9 [root@kuangshen home]# rm -f f1
10 [root@kuangshen home]# cat f2
11 I am f1 file
12 [root@kuangshen home]# cat f3
13 cat: f3: No such file or directory
```

通过上面的测试可以看出：当删除原始文件 f1 后，硬连接 f2 不受影响，但是符号连接 f1 文件无效；

依此您可以做一些相关的测试，可以得到以下全部结论：

删除符号连接f3,对f1,f2无影响；

删除硬连接f2，对f1,f3也无影响；

删除原文件f1，对硬连接f2没有影响，导致符号连接f3失效；

同时删除原文件f1,硬连接f2，整个文件会真正的被删除。

4、Vim的使用

4.1 什么是Vim编辑器

我自己使用之后的理解：其实就是一个文本编辑器查不到，用vim指令打开之后相当于去编辑记事本一样，然后用指令来选择输入模式，怎么保存，怎么推出啥的。

4.2 Vim的三种使用模式

基本上 vi/vim 共分为三种模式，分别是**命令模式 (Command mode)**，**输入模式 (Insert mode)** 和 **底线命令模式 (Last line mode)**。这三种模式的作用分别是：

命令模式：

用户刚刚启动 vi/vim，便进入了命令模式。

此状态下敲击键盘动作会被Vim识别为命令，而非输入字符。比如我们此时按下i，并不会输入一个字符，i被当作了一个命令。

以下是常用的几个命令：

- | | | |
|---|--|-------------------------|
| 1 | | i 切换到输入模式，以输入字符。 |
| 2 | | |
| 3 | | x 删除当前光标所在处的字符。 |
| 4 | | |
| 5 | | : 切换到底线命令模式，以在最底一行输入命令。 |

若想要编辑文本：启动Vim，进入了命令模式，按下i，切换到输入模式。

命令模式只有一些最基本的命令，因此仍要依靠底线命令模式输入更多命令。

输入模式：

在命令模式下按下i就进入了输入模式。

在输入模式中，可以使用以下按键：

- | | | |
|----|--|---------------------------------|
| 1 | | 字符按键以及Shift组合，输入字符 |
| 2 | | |
| 3 | | ENTER，回车键，换行 |
| 4 | | |
| 5 | | BACK SPACE，退格键，删除光标前一个字符 |
| 6 | | |
| 7 | | DEL，删除键，删除光标后一个字符 |
| 8 | | |
| 9 | | 方向键，在文本中移动光标 |
| 10 | | |
| 11 | | HOME/END，移动光标到行首/行尾 |
| 12 | | |
| 13 | | Page Up/Page Down，上/下翻页 |
| 14 | | |
| 15 | | Insert，切换光标为输入/替换模式，光标将变成竖线/下划线 |
| 16 | | |
| 17 | | ESC，退出输入模式，切换到命令模式 |

底线命令模式

在命令模式下按下: (英文冒号) 就进入了底线命令模式。

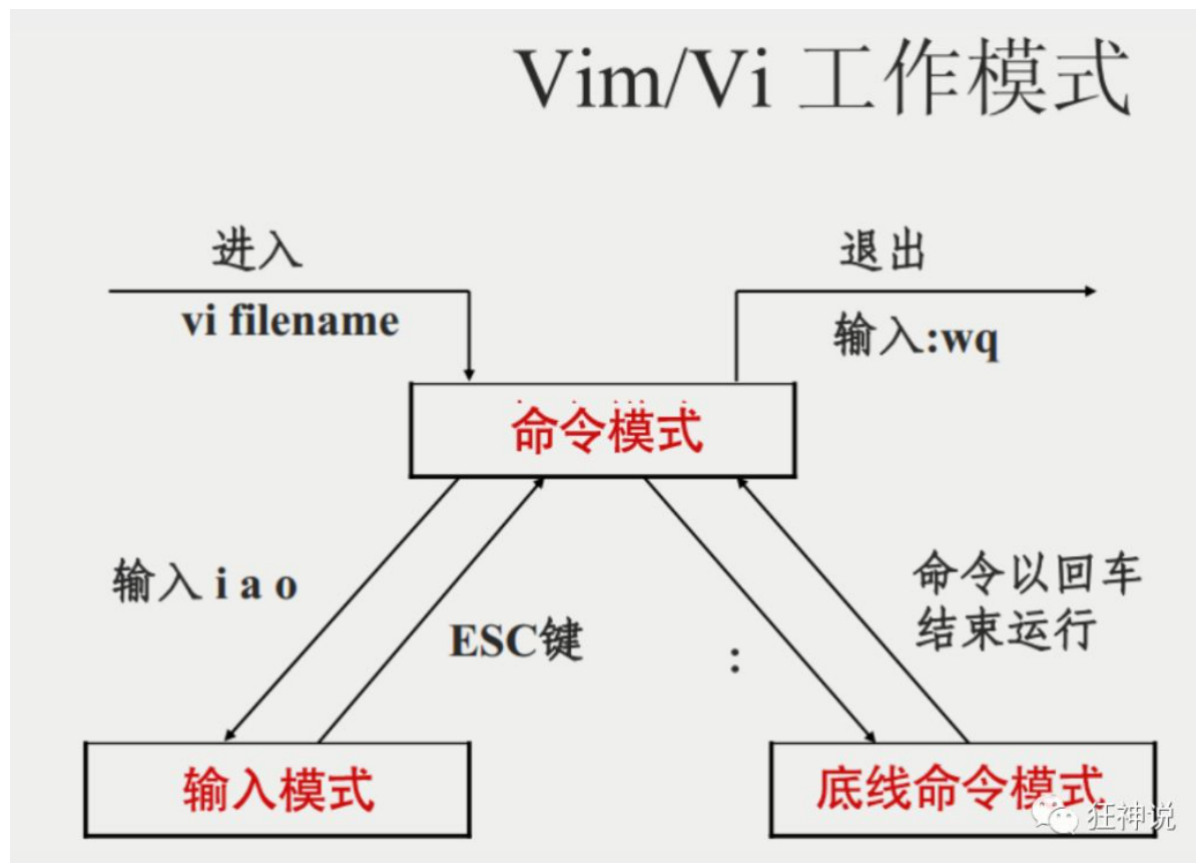
底线命令模式可以输入单个或多个字符的命令, 可用的命令非常多。

在底线命令模式中, 基本的命令有:

```
1 | :q 退出程序
2 |
3 | :w 保存文件
```

按ESC键可随时退出底线命令模式。

简单的说, 我们可以将这三个模式想成底下的图标来表示:



使用示例:

```
[root@llq ~]# cd home
[root@llq home]# ls
lilingqi  redis  www
[root@llq home]# cd lilingqi
[root@llq lilingqi]# vim llq
```


接下来输入i进入输入模式

[illegible]

按Esc退出输入模式，输入英文冒号:进入底线命令模式

[illegible]

: wq退出保存

查看llq内容:

```
[root@llq /]# cd home
[root@llq home]# ls
lilingqi  redis  www
[root@llq home]# cd lilingqi
[root@llq lilingqi]# vim llq
[root@llq lilingqi]# cat llq
hello vim
[root@llq lilingqi]#
```

4.3 Vim按键说明

第一部分：一般模式可用的光标移动、复制粘贴、搜索替换等

移动光标的方法	
h 或 向左箭头键(←)	光标向左移动一个字符
j 或 向下箭头键(↓)	光标向下移动一个字符
k 或 向上箭头键(↑)	光标向上移动一个字符
l 或 向右箭头键(→)	光标向右移动一个字符
[Ctrl] + [f]	屏幕『向下』移动一页，相当于 [Page Down]按键 (常用)
[Ctrl] + [b]	屏幕『向上』移动一页，相当于 [Page Up] 按键 (常用)
[Ctrl] + [d]	屏幕『向下』移动半页
[Ctrl] + [u]	屏幕『向上』移动半页
+	光标移动到非空格符的下一行
-	光标移动到非空格符的上一行
n< space>	那个 n 表示『数字』，例如 20 。按下数字后再按空格键，光标会向右移动这一行的 n 个字符。
0 或功能键[Home]	这是数字『0』：移动到这一行的最前面字符处 (常用)
\$ 或功能键[End]	移动到这一行的最后面字符处(常用)
H	光标移动到这个屏幕的最上方那一行的第一个字符
M	光标移动到这个屏幕的中央那一行的第一个字符
L	光标移动到这个屏幕的最下方那一行的第一个字符
G	移动到这个档案的最后一行(常用)
nG	n 为数字。移动到这个档案的第 n 行。例如 20G 则会移动到这个档案的第 20 行(可配合 :set nu)
gg	移动到这个档案的第一行，相当于 1G 啊! (常用)
n< Enter>	n 为数字。光标向下移动 n 行(常用)

搜索 替换	
/word	向光标之下寻找一个名称为 word 的字符串。例如要在档案内搜寻 vbird 这个字符串，就输入 /vbird 即可! (常用)
? word	向光标之上寻找一个字符串名称为 word 的字符串。
n	这个 n 是英文按键。代表重复前一个搜寻的动作。举例来说，如果刚刚我们执行 /vbird 去向下搜寻 vbird 这个字符串，则按下 n 后，会向下继续搜寻下一个名称为 vbird 的字符串。如果是执行 ?vbird 的话，那么按下 n 则会向上继续搜寻名称为 vbird 的字符串!
N	这个 N 是英文按键。与 n 刚好相反，为『反向』进行前一个搜寻动作。例如 /vbird 后，按下 N 则表示『向上』搜寻 vbird 。

删除、复制与粘贴	
x, X	在一行字当中，x 为向后删除一个字符 (相当于 [del] 按键)，X 为向前删除一个字符(相当于 [backspace] 亦即是退格键) (常用)
nx	n 为数字，连续向后删除 n 个字符。举例来说，我要连续删除 10 个字符，『10x』。
dd	删除光标所在的那一整行(常用)
ndd	n 为数字。删除光标所在的向下 n 行，例如 20dd 则是删除 20 行 (常用)
d1G	删除光标所在到第一行的所有数据
dG	删除光标所在到最后一行的所有数据
d\$	删除光标所在处，到该行的最后一个字符
d0	那个是数字的 0，删除光标所在处，到该行的最前面一个字符
yy	复制光标所在的那一行(常用)
nyy	n 为数字。复制光标所在的向下 n 行，例如 20yy 则是复制 20 行(常用)
y1G	复制光标所在行到第一行的所有数据
yG	复制光标所在行到最后一行的所有数据
y0	复制光标所在的那个字符到该行行首的所有数据
y\$	复制光标所在的那个字符到该行行尾的所有数据
p, P	p 为将已复制的数据在光标下一行贴上，P 则为贴在光标上一行！举例来说，我目前光标在第 20 行，且已经复制了 10 行数据。则按下 p 后，那 10 行数据会贴在原本的 20 行之后，亦即由 21 行开始贴。但如果是按下 P 呢？那么原本的 20 行会被推到变成 30 行。(常用)
J	将光标所在行与下一行的数据结合成同一行
c	重复删除多个数据，例如向下删除 10 行，[10cj]
u	复原前一个动作。(常用)
[Ctrl]+r	重做上一个动作。(常用)

第二部分：一般模式切换到编辑模式的可用的按钮说明

进入输入或取代的编辑模式	
i, I	进入输入模式(Insert mode): i 为『从目前光标所在处输入』，I 为『在目前所在行的第一个非空格符处开始输入』。(常用)
a, A	进入输入模式(Insert mode): a 为『从目前光标所在的下一个字符处开始输入』，A 为『从光标所在行的最后一个字符处开始输入』。(常用)
o, O	进入输入模式(Insert mode): 这是英文字母 o 的大小写。o 为『在目前光标所在的下一行处输入新的一行』；O 为在目前光标所在处的上一行输入新的一行！(常用)
r, R	进入取代模式(Replace mode): r 只会取代光标所在的那一个字符一次；R 会一直取代光标所在的文字，直到按下 ESC 为止；(常用)
[Esc]	退出编辑模式，回到一般模式中(常用)

S

第三部分：一般模式切换到指令行模式的可用的按钮说明

指令行的储存、离开等指令	
:w	将编辑的数据写入硬盘档案中(常用)
:w!	若文件属性为『只读』时，强制写入该档案。不过，到底能不能写入，还是跟你对该档案的档案权限有关啊！
:q	离开 vi (常用)
:q!	若曾修改过档案，又不想储存，使用！为强制离开不储存档案。
注意一下啊，那个惊叹号 (!) 在 vi 当中，常常具有『强制』的意思~	
:wq	储存后离开，若为 :wq! 则为强制储存后离开 (常用)
ZZ	这是大写的 Z 喔！若档案没有更动，则不储存离开，若档案已经被更动过，则储存后离开！
:w [filename]	将编辑的数据储存成另一个档案 (类似另存新档)
:r [filename]	在编辑的数据中，读入另一个档案的数据。亦即将『filename』这个档案内容加到光标所在行后面
:n1,n2 w [filename]	将 n1 到 n2 的内容储存成 filename 这个档案。
:! command	暂时离开 vi 到指令行模式下执行 command 的显示结果！例如『:! ls /home』即可在 vi 当中看 /home 底下以 ls 输出的档案信息！
:set nu	显示行号，设定之后，会在每一行的前缀显示该行的行号
:set nonu	与 set nu 相反，为取消行号！

5、账号管理

5.1 为什么需要账号管理

Linux系统是一个多用户多任务的分时操作系统，任何一个要使用系统资源的用户，都必须首先向系统管理员申请一个账号，然后以这个账号的身份进入系统。

用户的账号一方面可以帮助系统管理员对使用系统的用户进行跟踪，并控制他们对系统资源的访问；另一方面也可以帮助用户组织文件，并为用户提供安全性保护。

每个用户账号都拥有一个唯一的用户名和各自的口令。

用户在登录时键入正确的用户名和口令后，就能够进入系统和自己的主目录。

实现用户账号的管理，要完成的工作主要有如下几个方面：

用户账号的添加、删除与修改。

用户口令的管理。

用户组的管理。

5.2 用户的增删改以及切换

5.2.1 添加用户

语法：

```
1 | useradd 选项 用户名
```

参数：

```
1  -c comment 指定一段注释性描述。
2
3  -d 目录 指定用户主目录，如果此目录不存在，则同时使用-m选项，可以创建主目录。
4
5  -g 用户组 指定用户所属的用户组。
6
7  -G 用户组，用户组 指定用户所属的附加组。
8
9  -m 使用者目录如不存在则自动建立。
10
11 -s shell文件 指定用户的登录shell。
12
13 -u 用户号 指定用户的用户号，如果同时有-o选项，则可以重复使用其他用户的标识号。
```

示例：

```
1 [root@llq home]# ls
2 lilingqi redis www
3 [root@llq home]# useradd llq # 创建新用户llq
4 [root@llq home]# ls
5 lilingqi llq redis www
```

增加用户账号就是在/etc/passwd文件中为新用户增加一条记录，同时更新其他系统文件如/etc/shadow, /etc/group等。

5.2.2 linux下的切换用户

- 1.切换用户的命令为：su username 【username是你的用户名哦】
- 2.从普通用户切换到root用户，还可以使用命令：sudo su
- 3.在终端输入exit或logout或使用快捷方式ctrl+d，可以退回到原来用户，其实ctrl+d也是执行的exit命令
- 4.在切换用户时，如果想在切换用户之后使用新用户的工作环境，可以在su和username之间加-，例如：【su - root】

\$表示普通用户

#表示超级用户，也就是root用户

5.2.3 删除账户

如果一个用户的账号不再使用，可以从系统中删除。

删除用户账号就是要将/etc/passwd等系统文件中的该用户记录删除，必要时还删除用户的主目录。

语法：

```
1 userdel 选项 用户名
```

常用的选项是 -r，它的作用是把用户的主目录一起删除。

示例：

```
1 [root@llq home]# userdel lilingqi #这里没有删除主目录
2 [root@llq home]# ls
3 lilingqi llq redis www
4 [root@llq home]# userdel -r lilingqi
5 userdel: user 'lilingqi' does not exist
6 [root@llq home]# userdel -r llq #此命令删除用户kuangshen在系统文件中（主要是/etc/passwd, /etc/shadow, /etc/group等）的记录，同时删除用户的主目录。
7 [root@llq home]# ls
8 lilingqi redis www
```

5.2.4 修改用户

修改用户账号就是根据实际情况更改用户的有关属性，如用户号、主目录、用户组、登录Shell等。

修改已有用户的信息使用usermod命令。

语法：

```
1 usermod 选项 用户名
```

常用的参数选项包括-c, -d, -m, -g, -G, -s, -u以及-o等，这些选项的意义与useradd命令中的选项一样，可以为用户指定新的资源值

示例：

```
1 # usermod -s /bin/ksh -d /home/z -g developer lilingqi
```

此命令将用户lilingqi的登录Shell修改为ksh，主目录改为/home/z，用户组改为developer。

5.2.5 用户口令的管理

用户管理的一项重要内容是用户口令的管理。用户账号刚创建时没有口令，但是被系统锁定，无法使用，必须为其指定口令后才可以使用，即使是指定空口令。

指定和修改用户口令的Shell命令是passwd。超级用户可以为自己和其他用户指定口令，普通用户只能用它修改自己的口令。

语法：

```
1 passwd 选项 用户名
```

参数及其选项：

```
1 -l 锁定口令，即禁用账号。
2
3 -u 口令解锁。
4
5 -d 使账号无口令。
6
7 -f 强迫用户下次登录时修改口令。
```

如果默认用户名，则修改当前用户的口令。

例如，假设当前用户是lilingqi，则下面的命令修改该用户自己的口令：

```
1 $ passwd
2 old password:*****
3 New password:*****
4 Re-enter new password:*****
```

如果是超级用户，可以用下列形式指定任何用户的口令：

```
1 # passwd lilinqqi
2 New password:*****
3 Re-enter new password:*****
```

普通用户修改自己的口令时，passwd命令会先询问原口令，验证后再要求用户输入两遍新口令，如果两次输入的口令一致，则将这个口令指定给用户；而超级用户为用户指定口令时，就不需要知道原口令。

为用户指定空指令，执行下面的命令：

```
1 # passwd -d lilinqqi
```

此命令将用户 kuangshen的口令删除，这样用户 kuangshen下一次登录时，系统就不再允许该用户登录了。

passwd 命令还可以用 -l(lock) 选项锁定某一用户，使其不能登录，例如：

```
1 # passwd -l lilinqqi
```

6、用户组管理

每个用户都有一个用户组，系统可以对一个用户组中的所有用户进行集中管理。不同Linux 系统对用户组的规定有所不同，如Linux下的用户属于与它同名的用户组，这个用户组在创建用户时同时创建。

6.1 增加一个用户组

增加一个新的用户组使用groupadd命令

```
1 groupadd 选项 用户组
```

参数选项：

```
1 -g GID 指定新用户组的组标识号（GID）。
2
3 -o 一般与-g选项同时使用，表示新用户组的GID可以与系统已有用户组的GID相同。
```

示例：

```
1 # groupadd -g 101 group2 #此命令向系统中增加了一个新组group2，同时指定新组的组标识号是101。
```

6.2 删除一个用户组

如果要删除一个已有的用户组，使用groupdel命令：


```
1 | groupdel 用户组
```

示例：从系统中删除group1

```
1 | # groupdel group
```

6.3 修改用户组

语法：

```
1 | groupmod 选项 用户组
```

参数选项：

```
1 | -g GID 为用户组指定新的组标识号。
2 |
3 | -o 与-g选项同时使用，用户组的新GID可以与系统已有用户组的GID相同。
4 |
5 | -n新用户组 将用户组的名字改为新名字
```

示例：

```
1 | # 此命令将组group2的组标识号修改为102。
2 | groupmod -g 102 group2
3 |
4 | # 将组group2的标识号改为10000，组名修改为group3。
5 | groupmod -g 10000 -n group3 group2
```

6.4 切换组

如果一个用户同时属于多个用户组，那么用户可以在用户组之间切换，以便具有其他用户组的权限。

用户可以在登录后，使用命令newgrp切换到其他用户组，这个命令的参数就是目的用户组。例如：

```
1 | $ newgrp root
```

这条命令将当前用户切换到root用户组，前提条件是root用户组确实是该用户的主组或附加组。

6.4 跟用户信息相关的几个文件说明

完成用户管理的工作有许多种方法，但是每一种方法实际上都是对有关的系统文件进行修改。与用户和用户组相关的信息都存放在一些系统文件中，这些文件包括/etc/passwd, /etc/shadow, /etc/group等。下面分别介绍这些文件的内容。

6.4.1 /etc/passwd

Linux系统中的每个用户都在/etc/passwd文件中有一个对应的记录行，它记录了这个用户的一些基本属性。

这个文件对所有用户都是可读的。它的内容类似下面的例子：

```
1 # cat /etc/passwd
2
3 root:x:0:0:Superuser:/:
4 daemon:x:1:1:System daemons:/etc:
5 bin:x:2:2:Owner of system commands:/bin:
6 sys:x:3:3:Owner of system files:/usr/sys:
7 adm:x:4:4:System accounting:/usr/adm:
8 uucp:x:5:5:UUCP administrator:/usr/lib/uucp:
9 auth:x:7:21:Authentication administrator:/tcb/files/auth:
10 cron:x:9:16:Cron daemon:/usr/spool/cron:
11 listen:x:37:4:Network daemon:/usr/net/nls:
12 lp:x:71:18:Printer administrator:/usr/spool/lp:
```

从上面的例子我们可以看到，/etc/passwd中一行记录对应着一个用户，每行记录又被冒号(:)分隔为7个字段，其格式和具体含义如下：

```
1 | 用户名:口令:用户标识号:组标识号:注释性描述:主目录:登录Shell
```

1) "用户名"是代表用户账号的字符串。

通常长度不超过8个字符，并且由大小写字母和/或数字组成。登录名中不能有冒号(:)，因为冒号在这里是分隔符。

为了兼容起见，登录名中最好不要包含点字符(.)，并且不使用连字符(-)和加号(+)打头。

2) "口令"一些系统中，存放着加密后的用户口令字。

虽然这个字段存放的只是用户口令的加密串，不是明文，但是由于/etc/passwd文件对所有用户都可读，所以这仍是一个安全隐患。因此，现在许多Linux系统（如SVR4）都使用了shadow技术，把真正的加密后的用户口令字存放到/etc/shadow文件中，而在/etc/passwd文件的口令字段中只存放一个特殊的字符，例如"x"或者"*"。

3) "用户标识号"是一个整数，系统内部用它来标识用户。

一般情况下它与用户名是一一对应的。如果几个用户名对应的用户标识号是一样的，系统内部将把它们视为同一个用户，但是它们可以有不同的口令、不同的主目录以及不同的登录Shell等。

通常用户标识号的取值范围是0~65535。0是超级用户root的标识号，1~99由系统保留，作为管理账号，普通用户的标识号从100开始。在Linux系统中，这个界限是500。

4) "组标识号"字段记录的是用户所属的用户组。

它对应着/etc/group文件中的一条记录。

5) "注释性描述"字段记录着用户的一些个人情况。

例如用户的真实姓名、电话、地址等，这个字段并没有什么实际的用途。在不同的Linux系统中，这个字段的格式并没有统一。在许多Linux系统中，这个字段存放的是一段任意的注释性描述文字，用作finger命令的输出。

6) "主目录"，也就是用户的起始工作目录。

它是用户在登录到系统之后所处的目录。在大多数系统中，各用户的主目录都被组织在同一个特定的目录下，而用户主目录的名称就是该用户的登录名。各用户对自己的主目录有读、写、执行（搜索）权限，其他用户对此目录的访问权限则根据具体情况设置。

7) 用户登录后，要启动一个进程，负责将用户的操作传给内核，这个进程是用户登录到系统后运行的命令解释器或某个特定的程序，即Shell。

Shell是用户与Linux系统之间的接口。Linux的Shell有许多种，每种都有不同的特点。常用的有sh(Bourne Shell), csh(C Shell), ksh(Korn Shell), tcsh(TENEX/TOPS-20 type C Shell), bash(Bourne Again Shell)等。

系统管理员可以根据系统情况和用户习惯为用户指定某个Shell。如果不指定Shell，那么系统使用sh为默认的登录Shell，即这个字段的值为/bin/sh。

用户的登录Shell也可以指定为某个特定的程序（此程序不是一个命令解释器）。

利用这一特点，我们可以限制用户只能运行指定的应用程序，在该应用程序运行结束后，用户就自动退出了系统。有些Linux 系统要求只有那些在系统中登记了的程序才能出现在这个字段中。

8)系统中有一类用户称为伪用户（pseudo users）。

这些用户在/etc/passwd文件中也占有一条记录，但是不能登录，因为它们的登录Shell为空。它们的存在主要是方便系统管理，满足相应的系统进程对文件属主的要求。

常见的伪用户如下所示：

```
1 伪用户含义
2 bin 拥有可执行的用户命令文件
3 sys 拥有系统文件
4 adm 拥有帐户文件
5 uucp UUCP使用
6 lp lp或lpd子系统使用
7 nobody NFS使用
```

6.4.2 /etc/shadow

1、除了上面列出的伪用户外，还有许多标准的伪用户，例如：audit, cron, mail, usenet等，它们也都各自为相关的进程和文件所需要。

由于/etc/passwd文件是所有用户都可读的，如果用户的密码太简单或规律比较明显的话，一台普通的计算机就能够很容易地将它破解，因此对安全性要求较高的Linux系统都把加密后的口令字分离出来，单独存放在一个文件中，这个文件是/etc/shadow文件。有超级用户才拥有该文件读权限，这就保证了用户密码的安全性。

2、/etc/shadow中的记录行与/etc/passwd中的一一对应，它由pwconv命令根据/etc/passwd中的数据自动产生

它的文件格式与/etc/passwd类似，由若干个字段组成，字段之间用":"隔开。这些字段是：

```
1 登录名:加密口令:最后一次修改时间:最小时间间隔:最大时间间隔:警告时间:不活动时间:失效时间:标志
```

```
1 "登录名"是与/etc/passwd文件中的登录名相一致的用户账号
2
3 "口令"字段存放的是加密后的用户口令字，长度为13个字符。如果为空，则对应用户没有口令，登录时不需要口令；如果含有不属于集合 { ./0-9A-Za-z }中的字符，则对应的用户不能登录。
4
5 "最后一次修改时间"表示的是从某个时刻起，到用户最后一次修改口令时的天数。时间起点对不同的系统可能不一样。例如在SCO Linux 中，这个时间起点是1970年1月1日。
6
7 "最小时间间隔"指的是两次修改口令之间所需的最小天数。
8
9 "最大时间间隔"指的是口令保持有效的最大天数。
10
```

- 11 "警告时间"字段表示的是从系统开始警告用户到用户密码正式失效之间的天数。
- 12
- 13 "不活动时间"表示的是用户没有登录活动但账号仍能保持有效的最大天数。
- 14
- 15 "失效时间"字段给出的是一个绝对的天数，如果使用了这个字段，那么就给出相应账号的生存期。期满后，该账号就不再是一个合法的账号，也就不能再用来登录了。

6.4.3 /etc/group

用户组的所有信息都存放在/etc/group文件中。

将用户分组是Linux 系统中对用户进行管理 & 控制访问权限的一种手段。

每个用户都属于某个用户组；一个组中可以有多个用户，一个用户也可以属于不同的组。

当一个用户同时是多个组中的成员时，在/etc/passwd文件中记录的是用户所属的主组，也就是登录时所属的默认组，而其他组称为附加组。

用户要访问属于附加组的文件时，必须首先使用newgrp命令使自己成为所要访问的组中的成员。

用户组的所有信息都存放在/etc/group文件中。此文件的格式也类似于/etc/passwd文件，由冒号(:)隔开若干个字段，这些字段有：

1 组名:口令:组标识号:组内用户列表

- 1 "组名"是用户组的名称，由字母或数字构成。与/etc/passwd中的登录名一样，组名不应重复。
- 2
- 3 "口令"字段存放的是用户组加密后的口令字。一般Linux 系统的用户组都没有口令，即这个字段一般为空，或者是*。
- 4
- 5 "组标识号"与用户标识号类似，也是一个整数，被系统内部用来标识组。
- 6
- 7 "组内用户列表"是属于这个组的所有用户的列表/b]，不同用户之间用逗号(,)分隔。这个用户组可能是用户的主组，也可能是附加组。

7、磁盘管理

7.1 概述和指令

Linux磁盘管理好坏直接关系到整个系统的性能问题。

Linux磁盘管理常用命令为 df、du。

- df：列出文件系统的整体磁盘使用量
- du：检查磁盘空间使用量

df命令参数功能：检查文件系统的磁盘空间占用情况。可以利用该命令来获取硬盘被占用了多少空间，目前还剩下多少空间等信息。

语法：

1 df [-ahikHtm] [目录或文件名]

参数和选项

```

1  -a : 列出所有的文件系统，包括系统特有的 /proc 等文件系统；
2
3  -k : 以 KBytes 的容量显示各文件系统；
4
5  -m : 以 MBytes 的容量显示各文件系统；
6
7  -h : 以人们较易阅读的 GBytes, MBytes, KBytes 等格式自行显示；
8
9  -H : 以 M=1000K 取代 M=1024K 的进位方式；
10
11 -T : 显示文件系统类型，连同该 partition 的 filesystem 名称（例如 ext3）也列出；
12
13 -i : 不用硬盘容量，而以 inode 的数量来显示

```

示例:

```

1  # 将系统内所有的文件系统列出来！
2  # 在 Linux 底下如果 df 没有加任何选项
3  # 那么默认会将系统内所有的（不含特殊内存内的文件系统与 swap）都以 1 kbytes 的容量来列出来！
4  [root@llq /]# df
5  Filesystem      1k-blocks    Used Available Use% Mounted on
6  devtmpfs         889100        0    889100   0% /dev
7  tmpfs            899460       704    898756   1% /dev/shm
8  tmpfs            899460       496    898964   1% /run
9  tmpfs            899460        0    899460   0% /sys/fs/cgroup
10 /dev/vda1        41152812 6586736 32662368 17% /
11 tmpfs            179896        0    179896   0% /run/user/0

```

```

1  # 将容量结果以易读的容量格式显示出来
2  [root@kuangshen /]# df -h
3  Filesystem      Size  Used Avail Use% Mounted on
4  devtmpfs        869M    0  869M   0% /dev
5  tmpfs           879M  708K  878M   1% /dev/shm
6  tmpfs           879M  496K  878M   1% /run
7  tmpfs           879M    0  879M   0% /sys/fs/cgroup
8  /dev/vda1       40G   6.3G   32G  17% /
9  tmpfs           176M    0  176M   0% /run/user/0

```

```

1  # 将系统内的所有特殊文件格式及名称都列出来
2  [root@kuangshen /]# df -at
3  Filesystem      Type      1k-blocks    Used Available Use% Mounted on
4  sysfs           sysfs          0         0         0    - /sys
5  proc            proc          0         0         0    - /proc
6  devtmpfs        devtmpfs    889100        0    889100   0% /dev
7  securityfs      securityfs   0         0         0    -
  /sys/kernel/security
8  tmpfs           tmpfs      899460       708    898752   1% /dev/shm
9  devpts          devpts       0         0         0    - /dev/pts
10 tmpfs           tmpfs      899460       496    898964   1% /run
11 tmpfs           tmpfs      899460        0    899460   0% /sys/fs/cgroup
12 cgroup          cgroup       0         0         0    -
  /sys/fs/cgroup/systemd
13 pstore          pstore       0         0         0    - /sys/fs/pstore

```

14	cgroup	cgroup	0	0	0	-
	/sys/fs/cgroup/freezer					
15	cgroup	cgroup	0	0	0	-
	/sys/fs/cgroup/cpuset					
16	cgroup	cgroup	0	0	0	-
	/sys/fs/cgroup/hugetlb					
17	cgroup	cgroup	0	0	0	-
	/sys/fs/cgroup/blkio					
18	cgroup	cgroup	0	0	0	-
	/sys/fs/cgroup/net_cls,net_prio					
19	cgroup	cgroup	0	0	0	-
	/sys/fs/cgroup/memory					
20	cgroup	cgroup	0	0	0	-
	/sys/fs/cgroup/pids					
21	cgroup	cgroup	0	0	0	-
	/sys/fs/cgroup/cpu,cpuacct					
22	cgroup	cgroup	0	0	0	-
	/sys/fs/cgroup/devices					
23	cgroup	cgroup	0	0	0	-
	/sys/fs/cgroup/perf_event					
24	configfs	configfs	0	0	0	-
	/sys/kernel/config					
25	/dev/vda1	ext4	41152812	6586748	32662356	17% /
26	systemd-1	-	-	-	-	-
	/proc/sys/fs/binfmt_misc					
27	mqueue	mqueue	0	0	0	- /dev/mqueue
28	debugfs	debugfs	0	0	0	-
	/sys/kernel/debug					
29	hugetlbfs	hugetlbfs	0	0	0	- /dev/hugepages
30	tmpfs	tmpfs	179896	0	179896	0% /run/user/0
31	binfmt_misc	binfmt_misc	0	0	0	-
	/proc/sys/fs/binfmt_misc					

```

1 # 将 /etc 底下的可用的磁盘容量以易读的容量格式显示
2 [root@kuangshen ~]# df -h /etc
3 Filesystem      Size  Used Avail Use% Mounted on
4 /dev/vda1        40G   6.3G   32G   17% /

```

Linux du命令也是查看使用空间的，但是与df命令不同的是Linux du命令是对文件和目录磁盘使用的空间的查看，还是和df命令有一些区别的。

选项和参数：

```

1 -a : 列出所有的文件与目录容量，因为默认仅统计目录底下的文件量而已。
2
3 -h : 以人们较易读的容量格式（G/M）显示；
4
5 -s : 列出总量而已，而不列出每个各别的目录占用容量；
6
7 -S : 不包括子目录下的总计，与 -s 有点差别。
8
9 -k : 以 KBytes 列出容量显示；
10
11 -m : 以 MBytes 列出容量显示；

```

示例：

```

1 # 只列出当前目录下的所有文件夹容量（包括隐藏文件夹）：
2 # 直接输入 du 没有加任何选项时，则 du 会分析当前所在目录的文件与目录所占用的硬盘空间。
3 [root@llq home]# du
4 16  ./redis
5 8   ./www/.oracle_jre_usage # 包括隐藏文件的目录
6 24  ./www
7 48  . # 这个目录(.)所占用的总量

```

```

1 # 将文件的容量也列出来
2 [root@kuangshen home]# du -a
3 4   ./redis/.bash_profile
4 4   ./redis/.bash_logout
5 ....中间省略....
6 4   ./kuangstudy.txt # 有文件的列表了
7 48  .

```

```

1 # 检查根目录底下每个目录所占用的容量
2 [root@kuangshen home]# du -sm /*
3 0   /bin
4 146 /boot
5 ....中间省略....
6 0   /proc
7 ....中间省略....
8 1   /tmp
9 3026 /usr # 系统初期最大就是他了啦！
10 513 /var
11 2666 /www

```

通配符 * 来代表每个目录。

与 df 不一样的是，du 这个命令其实会直接到文件系统内去搜寻所有的文件数据。

7.2 磁盘的挂载和卸载

根文件系统之外的其他文件要想能够被访问，都必须通过“关联”至根文件系统上的某个目录来实现，此关联操作即为“挂载”，此目录即为“挂载点”，解除此关联关系的过程称之为“卸载”

Linux 的磁盘挂载使用 mount 命令，卸载使用 umount 命令。

挂载语法：

```
1 mount [-t 文件系统] [-L Label名] [-o 额外选项] [-n] 装置文件名 挂载点
```

示例：

```

1 # 将 /dev/hdc6 挂载到 /mnt/hdc6 上面！
2 [root@www ~]# mkdir /mnt/hdc6
3 [root@www ~]# mount /dev/hdc6 /mnt/hdc6
4 [root@www ~]# df
5 Filesystem            1k-blocks      Used Available Use% Mounted on
6 /dev/hdc6              1976312      42072   1833836    3% /mnt/hdc6

```

卸载语法：


```
1 | umount [-fn] 装置文件名或挂载点
```

选项和参数：

```
1 | -f : 强制卸除！可用在类似网络文件系统（NFS）无法读取到的情况下；  
2 |  
3 | -n : 不升级 /etc/mtab 情况下卸除。
```

卸载/dev/hdc6:

```
1 | [root@www ~]# umount /dev/hdc6
```

8、三种软件的安装方式

8.1 jdk安装（rpm安装）

1、rpm下载地址<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

2、如果有安装openjdk 则卸载

```
1 | [root@kuangshen ~]# java -version  
2 | java version "1.8.0_121"  
3 | Java(TM) SE Runtime Environment (build 1.8.0_121-b13)  
4 | Java HotSpot(TM) 64-Bit Server VM (build 25.121-b13, mixed mode)  
5 | # 检查  
6 | [root@kuangshen ~]# rpm -qa|grep jdk  
7 | jdk1.8.0_121-1.8.0_121-fcs.x86_64  
8 | # 卸载 -e --nodeps 强制删除  
9 | [root@kuangshen ~]# rpm -e --nodeps jdk1.8.0_121-1.8.0_121-fcs.x86_64  
10 | [root@kuangshen ~]# java -version  
11 | -bash: /usr/bin/java: No such file or directory # OK
```

3、安装DK

```
1 | # 安装java rpm  
2 | [root@kuangshen kuangshen]# rpm -ivh jdk-8u221-linux-x64.rpm  
3 |  
4 | # 安装完成后配置环境变量 文件: /etc/profile  
5 | JAVA_HOME=/usr/java/jdk1.8.0_221-amd64  
6 | CLASSPATH=%JAVA_HOME%/lib:%JAVA_HOME%/jre/lib  
7 | PATH=$PATH:$JAVA_HOME/bin:$JAVA_HOME/jre/bin  
8 | export PATH CLASSPATH JAVA_HOME  
9 | # 保存退出  
10 |  
11 | # 让新增的环境变量生效!  
12 | source /etc/profile  
13 | # 测试 java -version  
14 | [root@kuangshen java]# java -version  
15 | java version "1.8.0_221"  
16 | Java(TM) SE Runtime Environment (build 1.8.0_221-b11)  
17 | Java HotSpot(TM) 64-Bit Server VM (build 25.221-b11, mixed mode)
```

在阿里云服务器安装DK的时候，发现阿里云已经自动给安装好了

8.2 Tomcat的安装（解压缩安装）

- 1、安装好了Java环境后我们可以测试下Tomcat！准备好Tomcat的安装包！
- 2、将文件移动到/usr/tomcat/下，并解压！

```
1 [root@kuangshen kuangshen]# mv apache-tomcat-9.0.22.tar.gz /usr
2 [root@kuangshen kuangshen]# cd /usr
3 [root@kuangshen usr]# ls
4 apache-tomcat-9.0.22.tar.gz
5 [root@kuangshen usr]# tar -zxvf apache-tomcat-9.0.22.tar.gz # 解压
```

- 3、进入bin目录，运行Tomcat，和我们以前在Windows下看的都是一样的

```
1 # 执行：startup.sh -->启动tomcat
2 # 执行：shutdown.sh -->关闭tomcat
3 ./startup.sh
4 ./shutdown.sh
```

tomcat是我用宝塔面板傻瓜式安装的，宝塔安装的应用默认安装在www下的server

8.2 Docker的安装（yum安装）

基于 CentOS 7 安装

1. 官网安装参考手册：<https://docs.docker.com/install/linux/docker-ce/centos/>
2. 确定你是CentOS7及以上版本

```
1 [root@192 Desktop]# cat /etc/redhat-release
2 CentOS Linux release 7.2.1511 (Core)
```

3. yum安装gcc相关（需要确保 虚拟机可以上外网）

```
1 yum -y install gcc
2 yum -y install gcc-c++
```

4. 卸载旧版本

```
1 yum -y remove docker docker-common docker-selinux docker-engine
2 # 官网版本
3 yum remove docker \
4     docker-client \
5     docker-client-latest \
6     docker-common \
7     docker-latest \
8     docker-latest-logrotate \
9     docker-logrotate \
10    docker-engine
```

5. 安装需要的软件包

```
1 yum install -y yum-utils device-mapper-persistent-data lvm2
```

6. 设置stable镜像仓库

```
1 # 错误
2 yum-config-manager --add-repo
  https://download.docker.com/linux/centos/docker-ce.repo
3 ## 报错
4 [Errno 14] curl#35 - TCP connection reset by peer
5 [Errno 12] curl#35 - Timeout
6
7 # 正确推荐使用国内的
8 yum-config-manager --add-repo http://mirrors.aliyun.com/docker-
  ce/linux/centos/docker-ce.repo
```

7. 更新yum软件包索引

```
1 yum makecache fast
```

8. 安装Docker CE

```
1 yum -y install docker-ce docker-ce-cli containerd.io
```

9. 启动docke

```
1 systemctl start docker
```

10. 示例:

```
1 docker version
2
3 docker run hello-world
4
5 docker images
```

8.3 开启防火墙端口

确保Linux的防火墙端口是开启的，如果是阿里云，需要保证阿里云的安全组策略是开放的！相应的软件才可以使用。

示例:

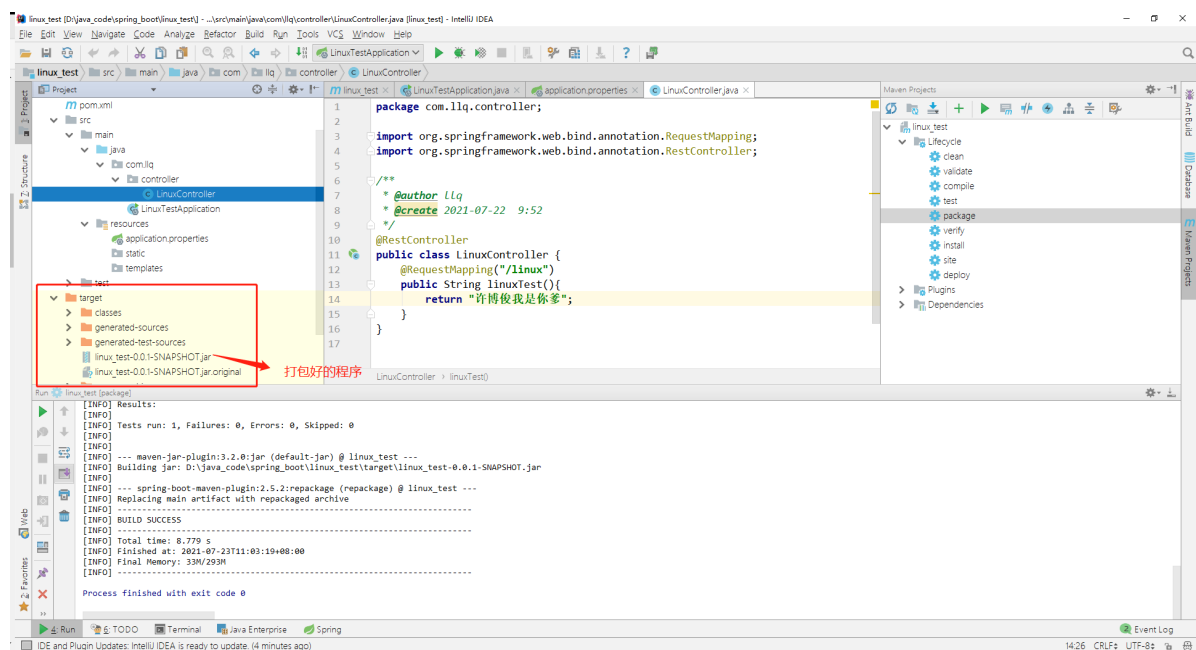
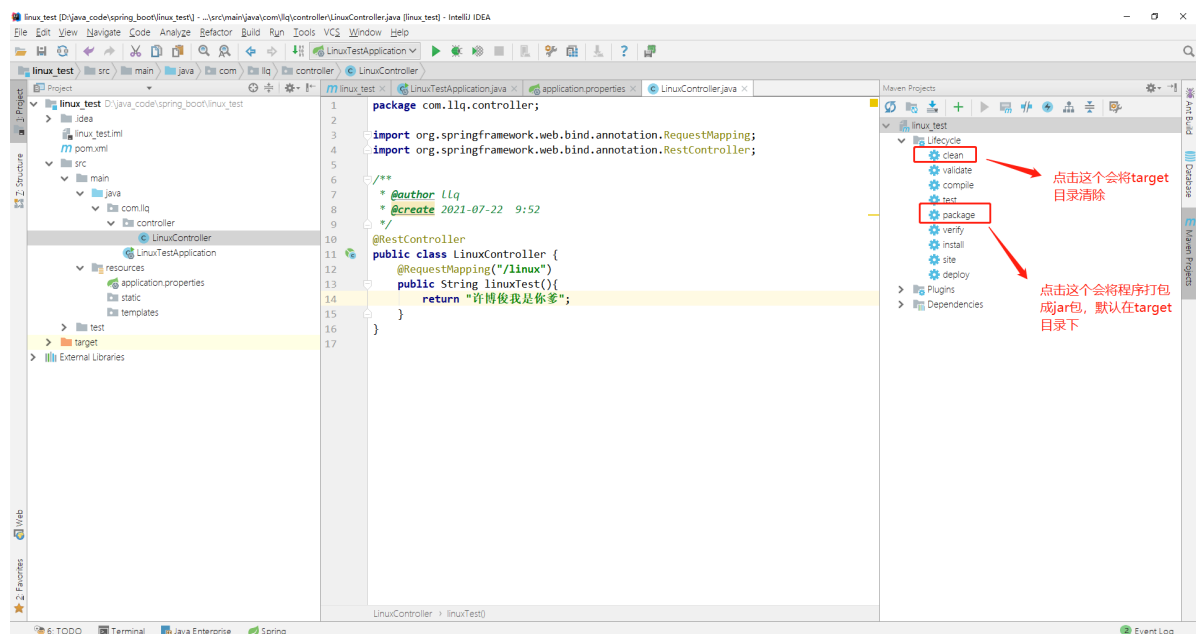
```
1 # 查看firewall服务状态
2 systemctl status firewalld
3
4 # 开启、重启、关闭、firewalld.service服务
5 # 开启
6 service firewalld start
7 # 重启
8 service firewalld restart
9 # 关闭
10 service firewalld stop
11
12 # 查看防火墙规则
13 firewall-cmd --list-all # 查看全部信息
14 firewall-cmd --list-ports # 只看端口信息
```

```
15
16 # 开启端口
17 端口命令: firewall-cmd --zone=public --add-port=80/tcp --permanent
18 重启防火墙: e'xi
19
20 命令含义:
21 --zone #作用域
22 --add-port=80/tcp #添加端口, 格式为: 端口/通讯协议
23 --permanent #永久生效, 没有此参数重启后失效
```

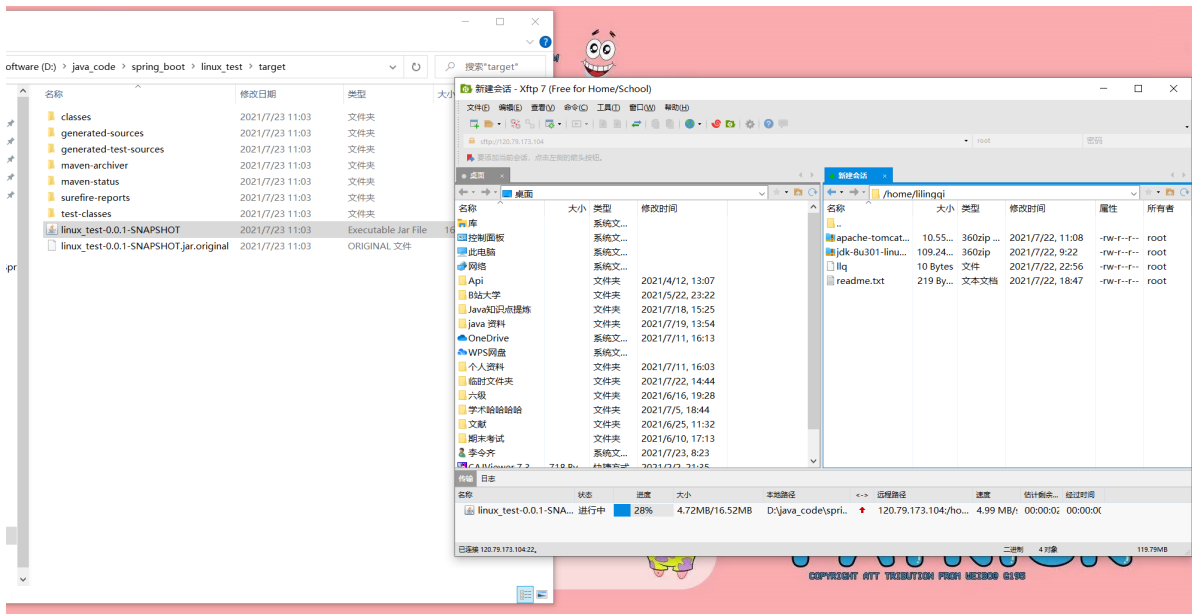
9.在云服务器上部署项目

9.1 部署springboot项目

步骤一：将要部署的项目打包

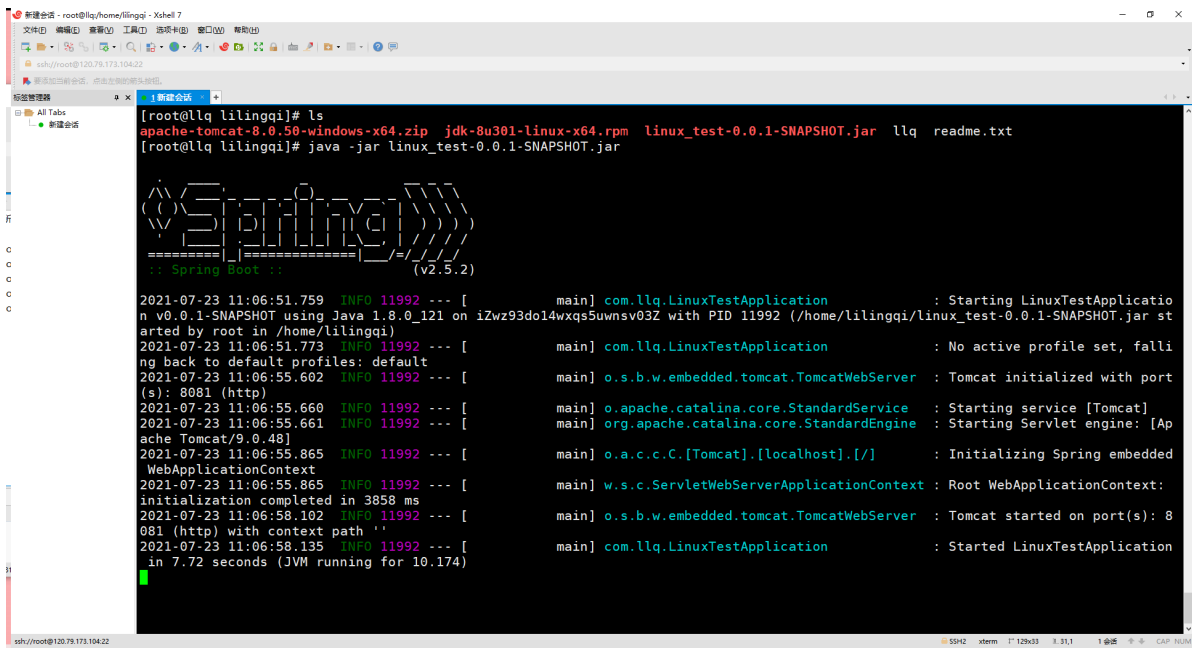


步骤二：打包好之后可以看到生成了一个target目录，下面的jar包就是我们打包好的项目，进入target目录将打包好的jar包用xftp上传到服务器。



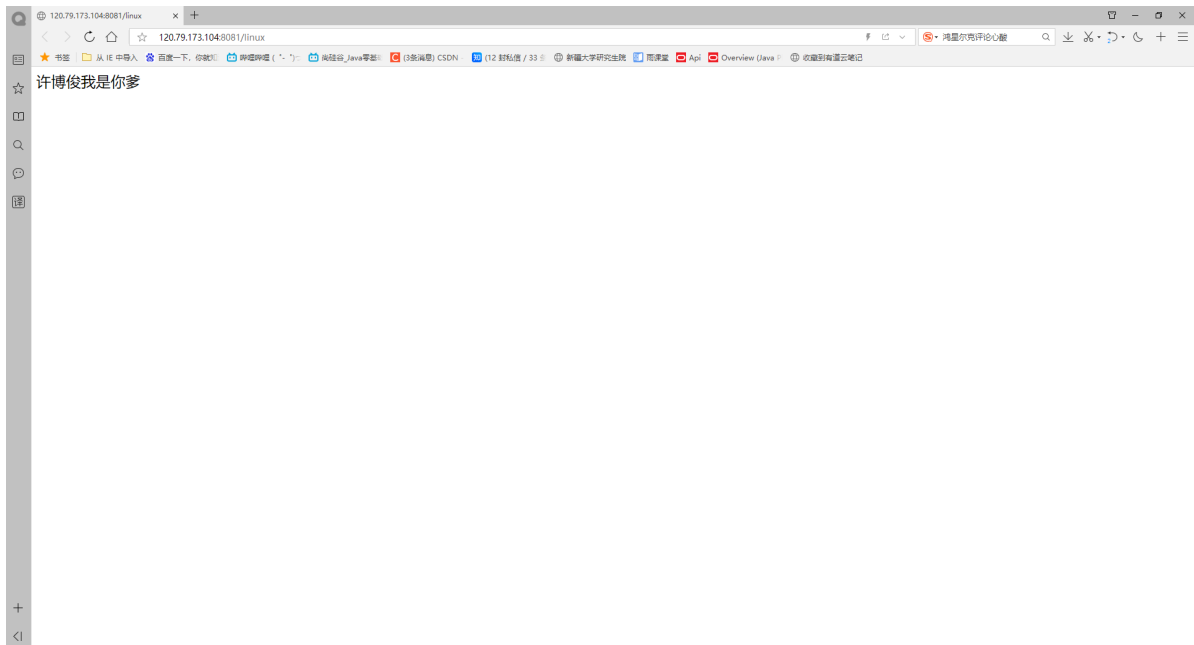
步骤三：在home/lilingqi 目录下运行下面指令：

```
1 [root@llq home]# cd lilingqi
2 [root@llq lilingqi]# ls
3 apache-tomcat-8.0.50-windows-x64.zip  jdk-8u301-linux-x64.rpm  linux_test-0.0.1-SNAPSHOT.jar  llq  readme.txt
4 [root@llq lilingqi]# java -jar linux_test-0.0.1-SNAPSHOT.jar #运行部署的项目
```



出现这个就是部署成功。

最后一步：用云服务器的公网ip地址加上端口号和路径来访问项目。



注意事项：

在部署项目的时候，一定要先保证项目在自己的服务器上能够跑成功，再去部署到云服务器上，除此之外要保证端口号是打开的！！！！！！

写在最后

这个Linux只能算是入门，其中怎么部署war项目到服务器还是不清楚，但是应该就是将项目打包放在tomcat的webapp下就可以访问了，还有怎么在本地服务器上去连接远程服务器也是不怎么会的，这些后期都是需要用到的。

本文参考文档：狂神linux笔记

参考视频链接：<https://www.bilibili.com/video/BV187411y7hF?p=15>